

1 1. A system for testing an enterprise system, comprising:
2 an aggregator for interfacing with an application under test that forms a part of an
3 enterprise application system;
4 a signal generator/database coupled to the aggregator for storing and retrieving
5 data; and
6 a plurality of probes each of which can be inserted between the aggregator and a
7 respective component of the application under test, the plurality of probes for recording
8 component data during a teach mode in the signal generator/database and injecting the
9 data stored in the signal generator/database into the enterprise system during a playback
10 mode to test the components.

1 2. The system according to claim 1, wherein the aggregator includes a graphical
2 interface for enabling a user to selectively insert the plurality of probes at various
3 locations in the application under test.

1 3. The system according to claim 2, wherein the graphical interface includes a
2 mechanism for selecting a breakpoint for playback mode.

1 4. The system according to claim 1, wherein the plurality of probes includes probes
2 for interfacing to components selected from the group consisting of databases, networks,
3 message queues, servlets, EJBs, legacy systems, and web servers.

1 5. The system according to claim 1, wherein at least one of the plurality of probes is
2 an EJB probe transparently inserted between an EJB client and an EJB.

1 6. The system according to claim 5, wherein a name proxy of the inserted EJB probe
2 and the EJB are manipulated to transparently insert the EJB probe.

1 7. The system according to claim 1, wherein the signal generator/database can store
2 component data selected from the group consisting of bean names, methods, arguments,
3 method ordering, transaction number, elapsed time, and object information.

1 8. The system according to claim 1, wherein the aggregator includes a graphical
2 interface having a mechanism to expand data associated with a component under test.

1 9. The system according to claim 8, wherein the graphical interface further includes
2 a mechanism to create a plurality of instances of the component under test and exercise
3 the component under test using data expanded from the data stored in the signal
4 generator/database.

1 10. A method for testing an enterprise system, comprising:
2 inserting a plurality of probes between an aggregator and respective components
3 of an application under test;
4 recording data received by the plurality of probes during a teach mode;
5 storing the recorded data in a database;
6 injecting the recorded data into the enterprise system during a playback mode;
7 recording data received by the plurality of probes during the playback mode; and
8 comparing actual and expected data.

1 11. The method according to claim 10, further including selecting a breakpoint
2 corresponding to a point associated with a component under test.

1 12. The method according to claim 11, further including running the application under
2 test until reaching the breakpoint and retrieving recorded data associated with the
3 component under test.

13. The method according to claim 12, further including expanding the data associated with the component under test and creating a plurality of instances of the component under test.

1 14. The method according to claim 13, further including load testing the component
2 under test with the expanded data.

1 15. The method according to claim 14, further including load testing the component
2 under test without compiling test code.

1 16. The method according to claim 11, further including selecting the component
2 under test from the group consisting of EJBs, web pages, web queues, databases, legacy
3 systems, and message queues.

1 17. The method according to claim 10, further including testing at least one of the
2 plurality of components in a transactional context.

1 18. The method according to claim 17, further including retrieving methods
2 associated with the at least one of the plurality of components in an order in which the
3 methods were called during the teach mode.

1 19. The method according to claim 10, further including transparently inserting an
2 EJB probe as one of the plurality of probes between an EJB client and an EJB component.

1 20. The method according to claim 19, further including transparently inserting the
2 EJB probe by replacing a name proxy of the EJB component with that of the EJB probe.

1 21. The method according to claim 19, further including using Java reflection to
2 generate the EJB probe from the EJB component.

1 22. The method according to claim 10, further including extracting execution time
2 associated with the plurality of probes.

1 23. A computer program product for testing an enterprise system comprising code for:
2 inserting a plurality of probes between an aggregator and respective components
3 of an application under test;
4 recording data received by the plurality of probes during a teach mode;
5 storing the recorded data in a database;
6 injecting the recorded data into the enterprise system during a playback mode;
7 recording data received by the plurality of probes during the playback mode; and
8 comparing actual and expected data.

1 24. The computer program product according to claim 23, further including code for
2 selecting a breakpoint corresponding to a point associated with a component under test.

1 25. The computer program product according to claim 24, further including code for
2 running the application under test until reaching the breakpoint and retrieving recorded
3 data associated with the component under test.

1 26. The computer program product according to claim 25, further including code for
2 expanding the data associated with the component under test and creating a plurality of
3 instances of the component under test.

1 27. The computer program product according to claim 26, further including code for
2 load testing the component under test with the expanded data.

1 28. The computer program product according to claim 23, further including code for
2 selecting the component under test from the group consisting of EJBs, web pages, EJBs,
3 web pages, web queues, databases, legacy systems, and message queues.

1 29. The computer program product according to claim 23, further including code for
2 load testing the component under test without compiling test code.

1 30. The computer program product according to claim 23, further including code for
2 testing at least one of the plurality of components in transactional context.

1 31. The computer program product according to claim 30, further including code for
2 retrieving methods associated with the at least one of the plurality of components in an
3 order in which the method were called during the teach mode.

1 32. The computer program product according to claim 23, further including code for
2 transparently inserting an EJB probe as one of the plurality of probes between an EJB
3 client and an EJB component.

1 33. The computer program product according to claim 32, further including code for
2 transparently inserting the EJB probe by replacing a proxy of the EJB component with
3 that of the EJB probe.

1 34. The computer program product according to claim 33, further including code for
2 using Java reflection to generate the EJB probe from the EJB component.

1 35. The computer program product according to claim 23, further including code for
2 extracting execution time associated with the plurality of probes.